Smarter Not Harder:

Scripting with Wireshark to Enhance Network Traffic Analysis

Nick Pelletier

Consultant

Mandiant

Agenda

- Wireshark
- Lua
- Why Not Both?
- Use Cases
- Advanced Uses
- Questions

Before We Begin

- This presentation:
  - Assumes very little
  - Is not meant to be exhaustive
  - Is starting point for projects
- Goals:
  - Spark ideas for custom use cases
  - Provide enough detail to get the ball rolling and the resources to build further
  - Reference Material

Wireshark

- Packet Analyzer
  - Free
  - Open Source
  - Cross Platform
- Dissectors
  - “Understand” various protocols
  - Human Readable Format
- Scriptable
  - Lua
  - API Access to Dissectors
- TShark

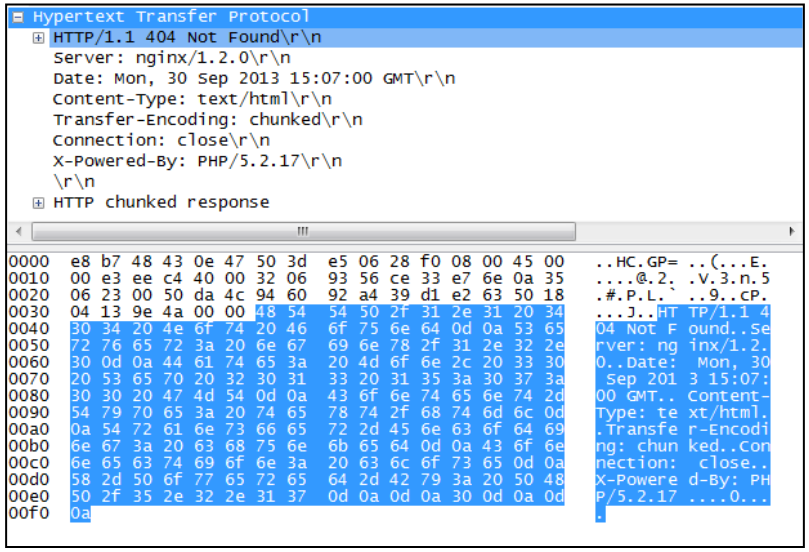
Wireshark - Dissectors

- Dissects the Protocol
- Great Variation
  - HTTP
  - AOL Instant Messenger
  - ZigBee
- Wireshark 1.8.2
  - 1,300+

Wireshark - Dissectors

- Example: HTTP
  - RFC2616
  - 176 Pages
- HTTP Dissector
  - 42 Fields
    - Not Including Higher Level
      - ip.addr
      - tcp.dst_port
  - Page Not Found?
    - Bytes 3F:41 == 0x343034
    - “http.response.code == 404”

“A correct cache MUST respond to a request with the most up-to-date response held by the cache that is appropriate to the request which meets one of the following conditions…”



Lua

- Lightweight Scripting Language
  - Very Often Embedded
  - C API
    - “Lua is especially useful for providing end users with an easy way to program the behavior of a software product without getting too far into its innards.”
- Extensible
  - Limited Native Capability
    - Flexible
    - Can Fit Many Models
    - Multi-Paradigm

Lua – Who Cares?

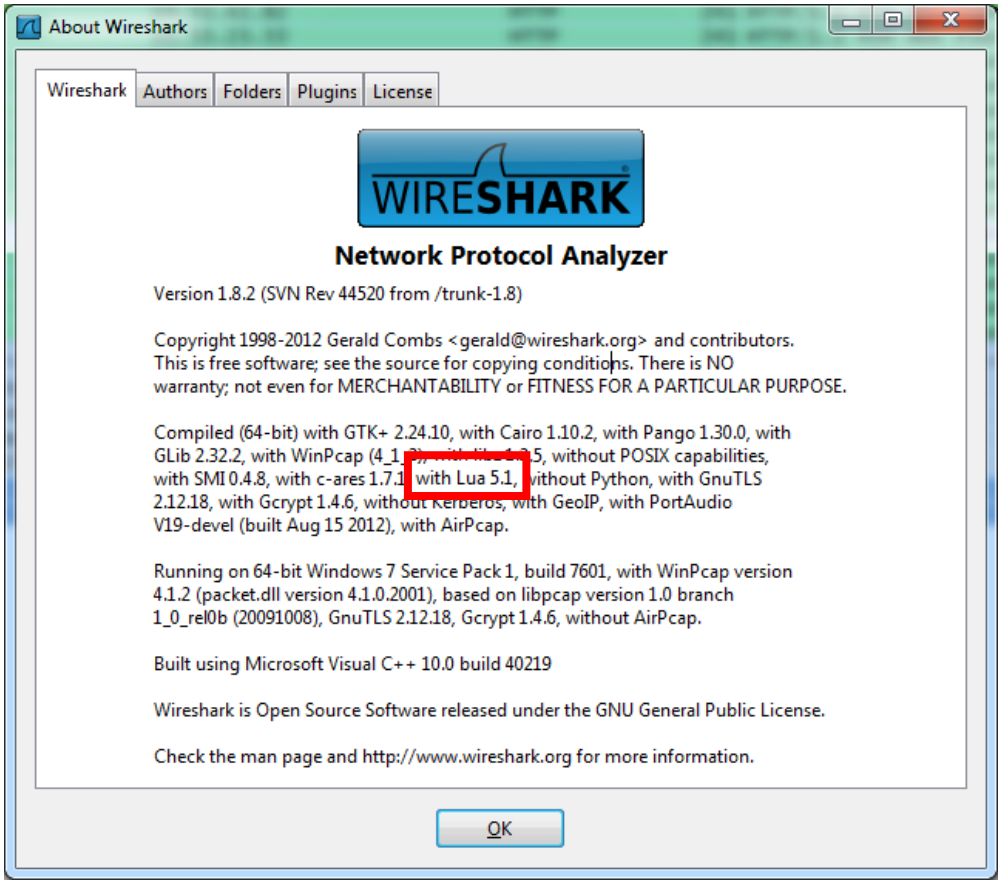
- Leverage Capabilities of Wireshark
  - Customize Output
    - Pseudo Narrative
  - Perform Specialized Tasks
    - Take Action on Specific Conditions
  - Development of Dissectors
    - Quick and Easy Dev With Lua
    - When Complete, Rewrite in C
- Don’t Reinvent the Wheel

Getting Setup

- Is Lua Supported?
  - Help > About
  - tshark -v
- Is Lua Enabled?
  - Wireshark install directory > init.lua

```
-- Set disable_lua to true to
--       disable Lua support.
disable_lua = false

if disable_lua then
    return
end
```


About Wireshark
Wireshark
Authors
Folders
Plugins
License
WIRESHARK
Network Protocol Analyzer
Version 1.8.2 (SVN Rev 44520 from /trunk-1.8)
Copyright 1998-2012 Gerald Combs <gerald@wireshark.org> and contributors.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
Compiled (64-bit) with GTK+ 2.24.10, with Cairo 1.10.2, with Pango 1.30.0, with
GLib 2.32.2, with WinPcap (4_1
with Lua 5.1,
2.12.18, with Gcrypt 1.4.6,
V19-devel (built Aug 15 2012), with AirPcap.
Running on 64-bit Windows 7 Service Pack 1, build 7601, with WinPcap version
4.1.2 (packet.dll version 4.1.0.2001), based on libpcap version 1.0 branch
1_0_rel0b (20091008), GnuTLS 2.12.18, Gcrypt 1.4.6, without AirPcap.
Built using Microsoft Visual C++ 10.0 build 40219
Wireshark is Open Source Software released under the GNU General Public License.
Check the man page and http://www.wireshark.org for more information.
OK

Script Basics – Listeners and Fields

- Listener
  - Defines Packets to Process
  - Initialized With Tap and (optional) Display Filter
- Field
  - Defines Dissected Data to Access
  - Initialized With Field Name

```
--Init tap
tap =       Listener.new("http","http.request")

--Define fields
host =      Field.new("http.host")
org_host =  Field.new("http.x_forwarded_for")
uri =       Field.new("http.request.uri")
```

Taps

- ip
- udp
- http
- bacapp
- h225
- actrace
- ansi_a
- ansi_map
- eth
- tcp
- wlan
- frame

Script Basics – Listeners and Fields

- Listener.packet()
  - Run each time a packet matches the Listener definition. In our example, every HTTP Request.
- Listener.draw()
  - Run periodically during execution. Used to redraw graphical components in Wireshark. Not used in our example.
- Listener.reset()
  - Run at the beginning and end of each capture run. Necessary for clearing any stored state between captures.

Assignment

Record the HTTP host for any POST request containing the term “login” in the URI

Myfirstscript.lua

```
--Init tap
tap =       Listener.new("http","http.request")

--Define fields
host =      Field.new("http.host")
uri =       Field.new("http.request.uri")
method =    Field.new("http.request.method")

function tap.packet(pinfo,tvb,tapdata)

    --If the request is a POST
    if tostring(method())=="POST" then

        --If the URI contains the string "login"
        if string.find(tostring(uri()),"login") ~= nil then

            --Print the host
            message(host())
        end

    end
end
```

Myfirstscript.lua

```
--Init tap
tap =       Listener.new("http","http.request")

--Define fields
host =      Field.new("http.host")
uri =       Field.new("http.request.uri")
method =    Field.new("http.request.method")

function tap.packet(pinfo,tvb,tapdata)

    --If the request is a POST
    if tostring(method())=="POST" then

        --If the URI contains the string "login"
        if string.find(tostring(uri()),"login") ~= nil then

            --Print the host
            message(host())
        end

    end
end
```

Myfirstscript.lua

```
--Init tap
tap =       Listener.new("http","http.request")

--Define fields
host =      Field.new("http.host")
uri =       Field.new("http.request.uri")
method =    Field.new("http.request.method")

function tap.packet(pinfo,tvb,tapdata)

    --If the request is a POST
    if tostring(method())=="POST" then

        --If the URI contains the string "login"
        if string.find(tostring(uri()),"login") ~= nil then

            --Print the host
            message(host())
        end

    end
end
```

Myfirstscript.lua

```
--Init tap
tap =       Listener.new("http","http.request")

--Define fields
host =      Field.new("http.host")
uri =       Field.new("http.request.uri")
method =    Field.new("http.request.method")

function tap.packet(pinfo,tvb,tapdata)

    --If the request is a POST
    if tostring(method())=="POST" then

        --If the URI contains the string "login"
        if string.find(tostring(uri()),"login") ~= nil then

            --Print the host
            message(host())
        end

    end
end
```

Myfirstscript.lua

```
--Init tap
tap =       Listener.new("http","http.request")

--Define fields
host =      Field.new("http.host")
uri =       Field.new("http.request.uri")
method =    Field.new("http.request.method")

function tap.packet(pinfo,tvb,tapdata)

    --If the request is a POST
    if tostring(method())=="POST" then

        --If the URI contains the string "login"
        if string.find(tostring(uri()),"login") ~= nil then

            --Print the host
            message(host())
        end

    end
end
```

Myfirstscript.lua

```
--Init tap
tap =       Listener.new("http","http.request")

--Define fields
host =      Field.new("http.host")
uri =       Field.new("http.request.uri")
method =    Field.new("http.request.method")

function tap.packet(pinfo,tvb,tapdata)

    --If the request is a POST
    if tostring(method())=="POST" then

        --If the URI contains the string "login"
        if string.find(tostring(uri()),"login") ~= nil then

            --Print the host
            message(host())
        end

    end
end
```

Myfirstscript.lua

```
--Init tap
tap =       Listener.new("http","http.request")

--Define fields
host =      Field.new("http.host")
uri =       Field.new("http.request.uri")
method =    Field.new("http.request.method")

function tap.packet(pinfo,tvb,tapdata)

    --If the request is a POST
    if tostring(method())=="POST" then

        --If the URI contains the string "login"
        if string.find(tostring(uri()),"login") ~= nil then

            --Print the host
            message(host())
        end

    end
end
```

Running the Script

- On-Demand (Command Line)
  - **`wireshark –X lua_script:script.lua`**
  - **`tshark –q –X lua_script:script.lua`**
- On-Load
  - Plugins Directory
    - **`%PROGRAMFILES%\Wireshark\[version]\`**
    - **`%APPDATA%\Roaming\Wireshark\plugins\`**
  - init.lua
    - **`dofile("C:\Path\To\Script.lua")`**

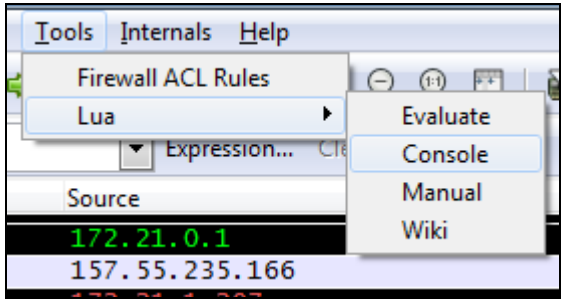
Running the Script

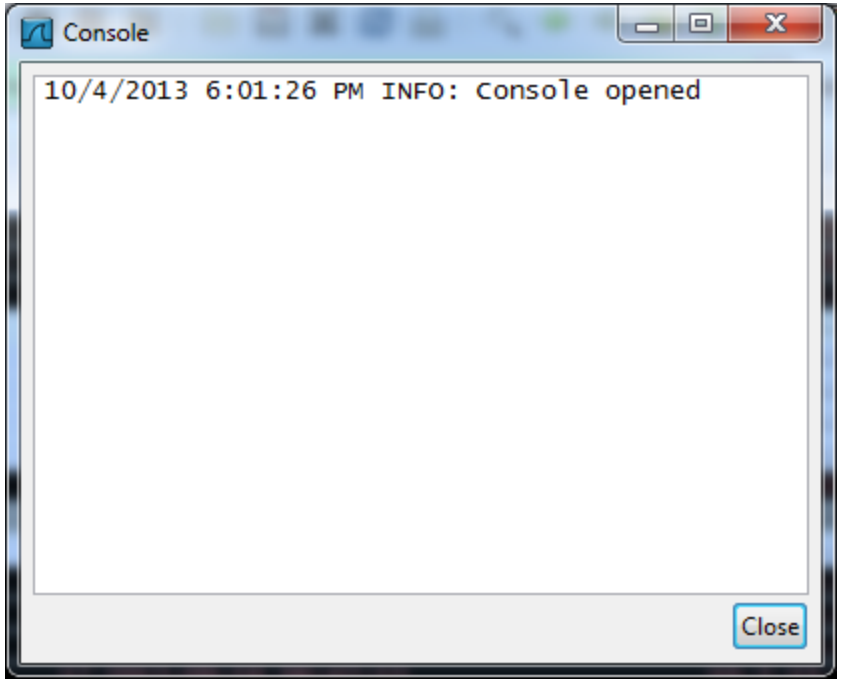
- Start Wireshark

```
wireshark –X lua_script:script.lua
```

Running the Script

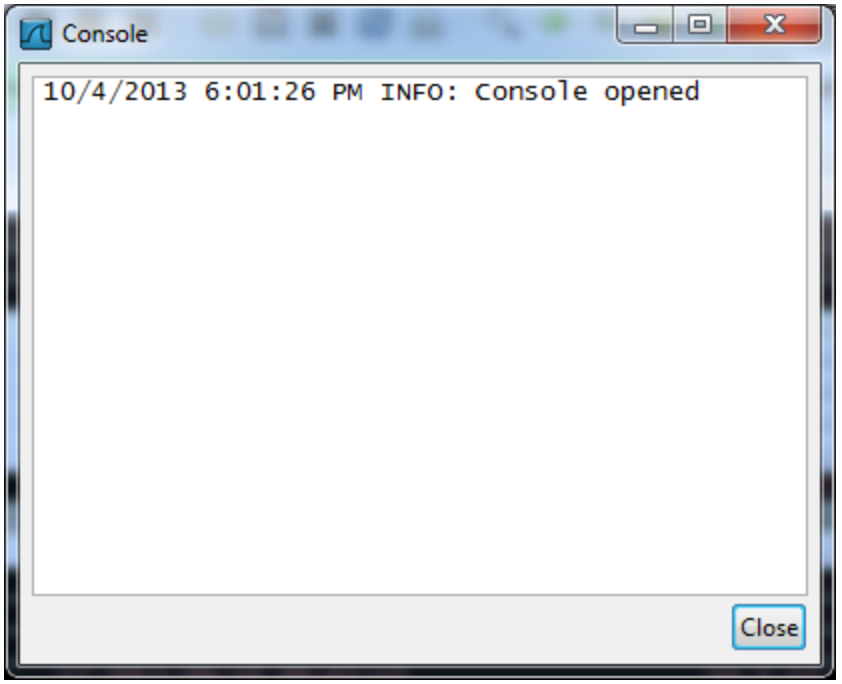
- Start Wireshark
- Open Lua Console





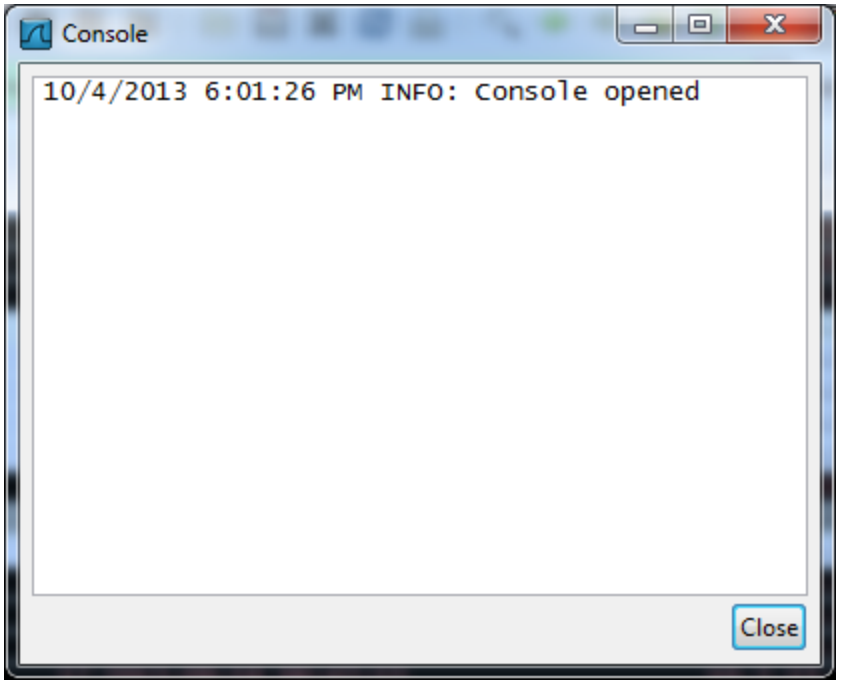
Running the Script

- Start Wireshark
- Open Lua Console
- Wait



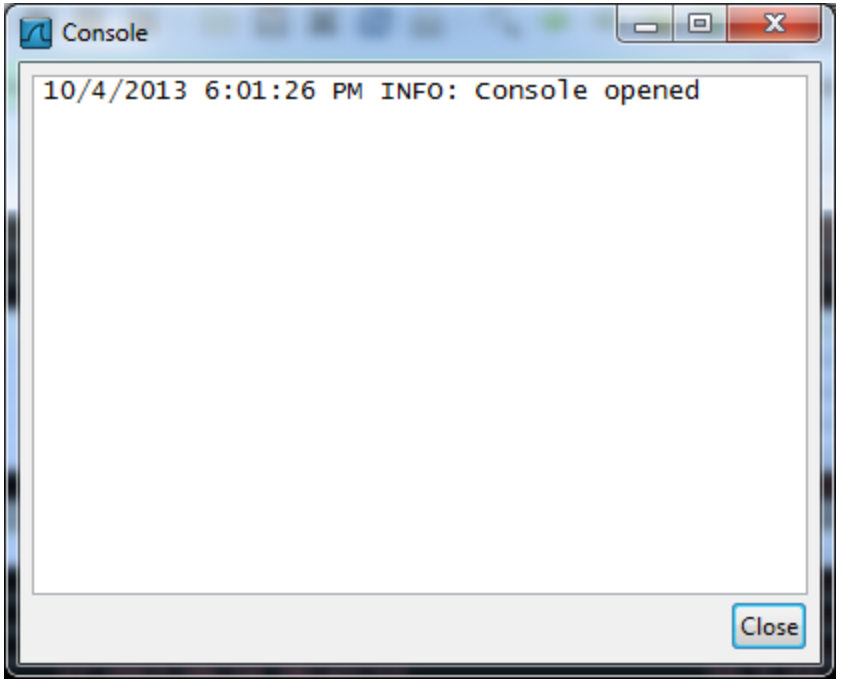
Running the Script

- Start Wireshark
- Open Lua Console
- Wait
- Wait



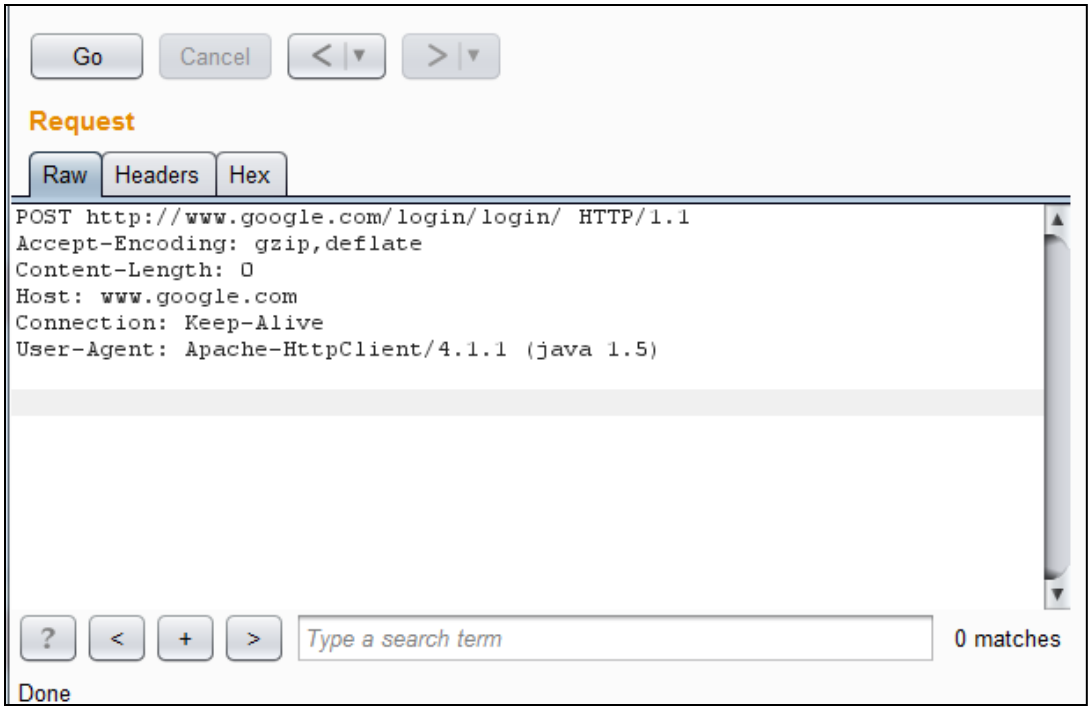
Running the Script

- Start Wireshark
- Open Lua Console
- Wait
- Wait
- Wait



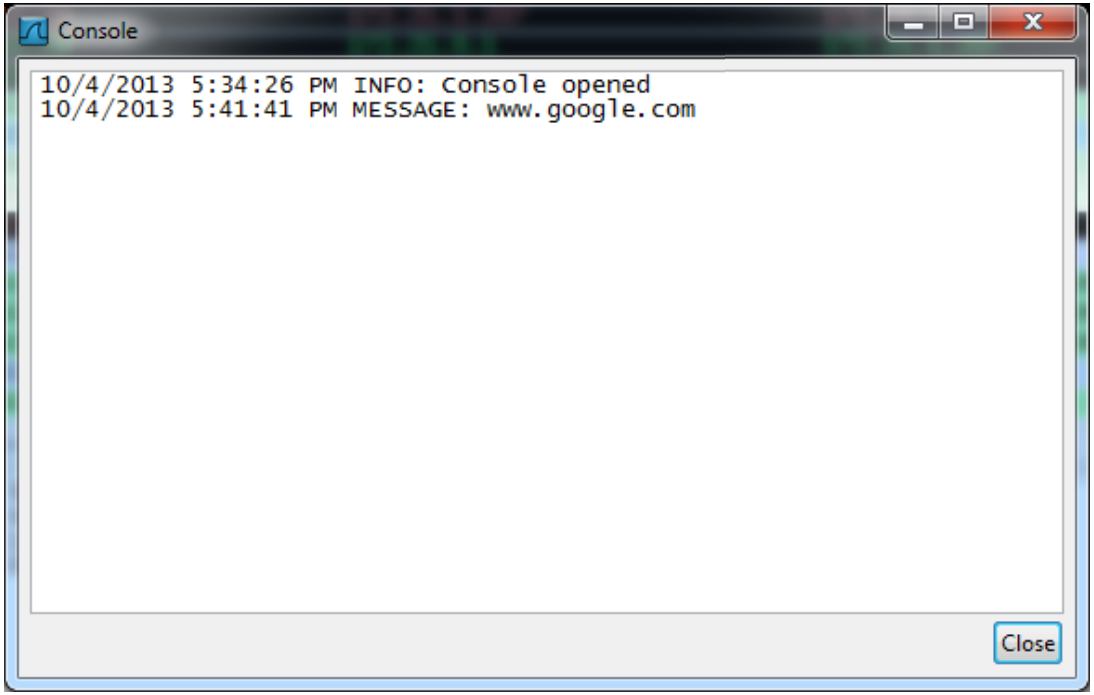
Running the Script

- Start Wireshark
- Open Lua Console
- Wait
- Wait
- Wait
- Cheat



Running the Script

- Start Wireshark
- Open Lua Console
- Wait
- Wait
- Wait
- Cheat
- Success!



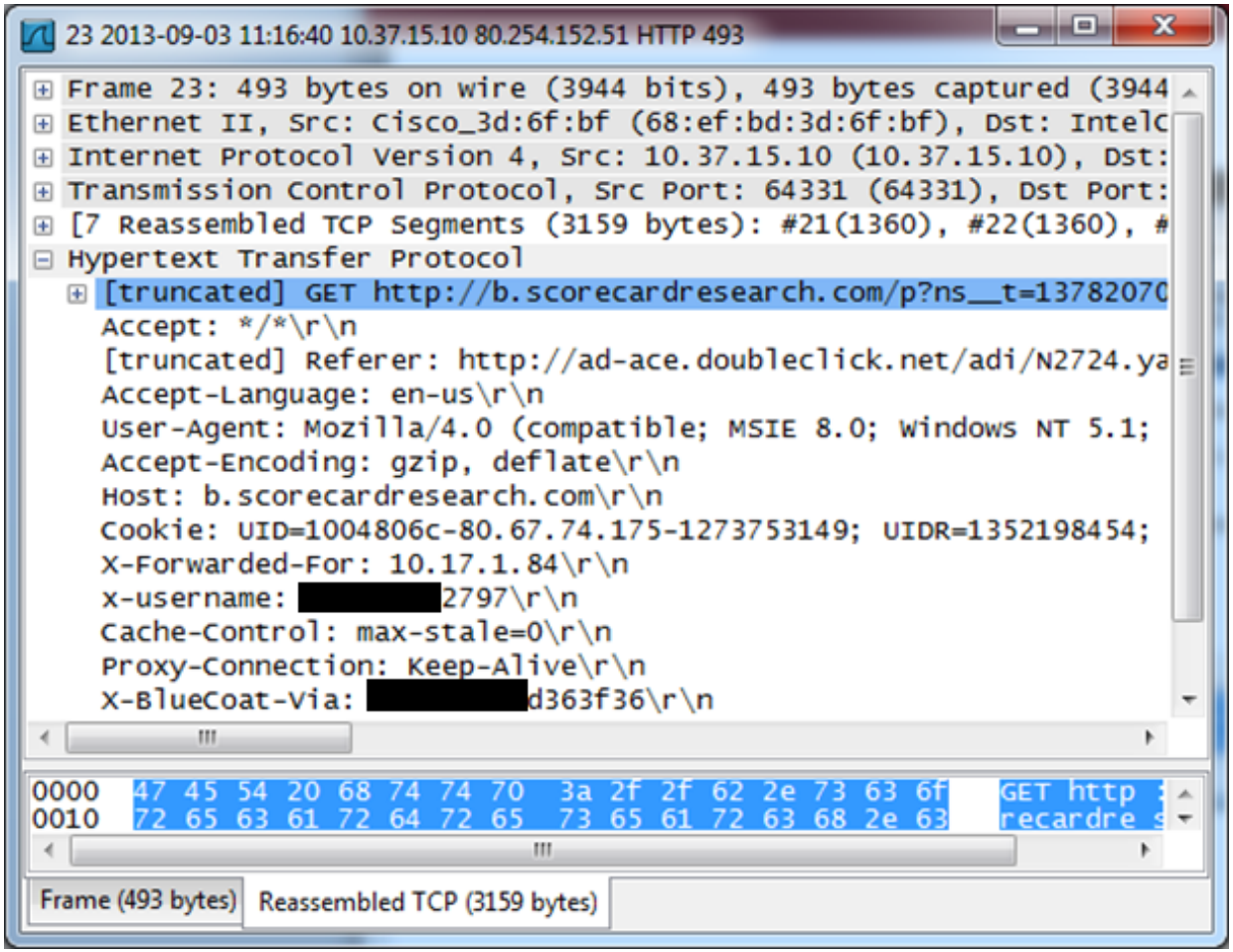
That's Nice, But Why?

- Display filters are great, but…
  - `http.request.full_uri contains "login"`
- Obtain "buried" protocol data
- Automate collection of information
- Dissecting custom traffic

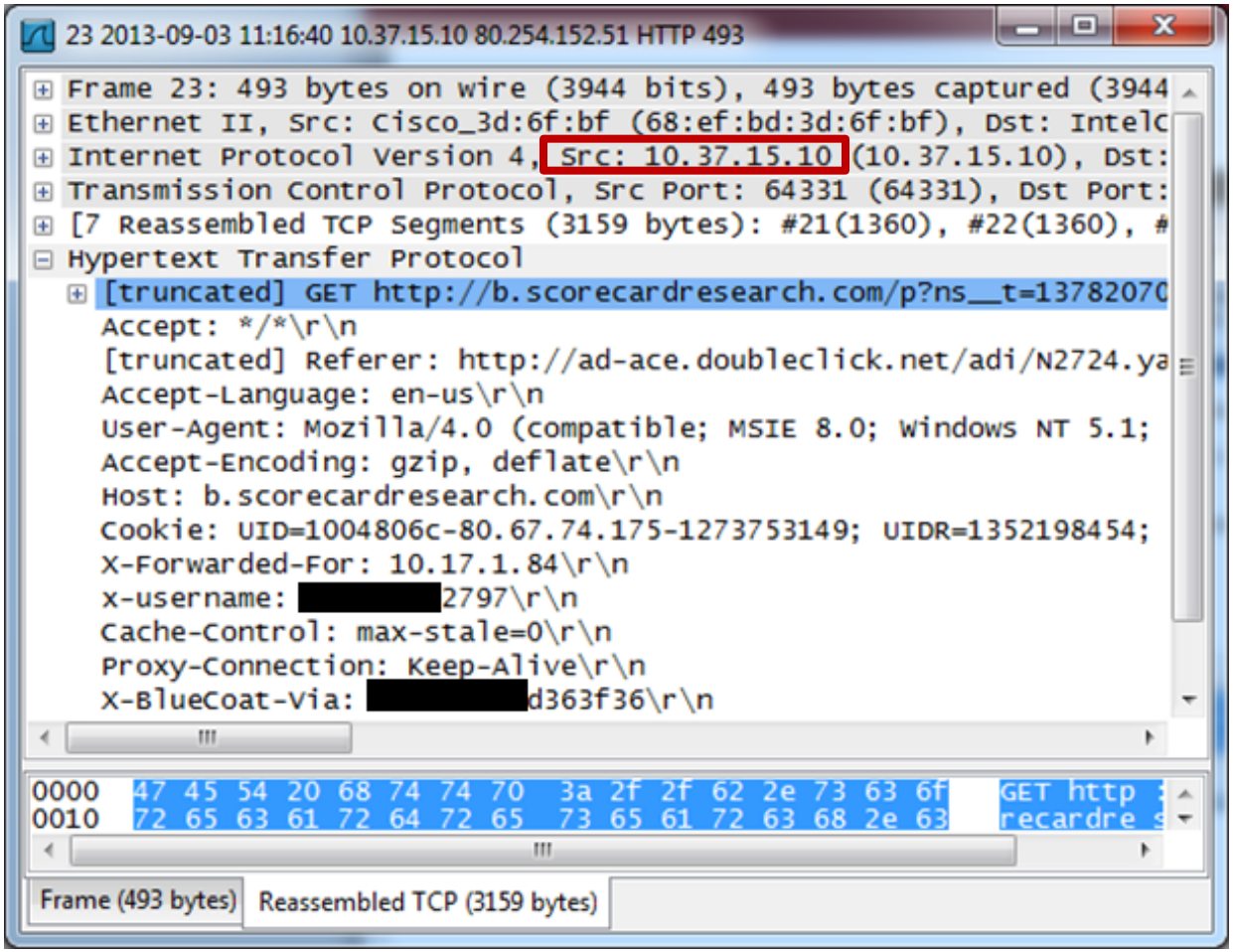
A More Practical Example

Scenario: You are tasked with monitoring a network for HTTP beacons to known network based indicators (NBIs). However, due to the configuration of the network, all IP traffic appears to be from a proxy.

A More Practical Example

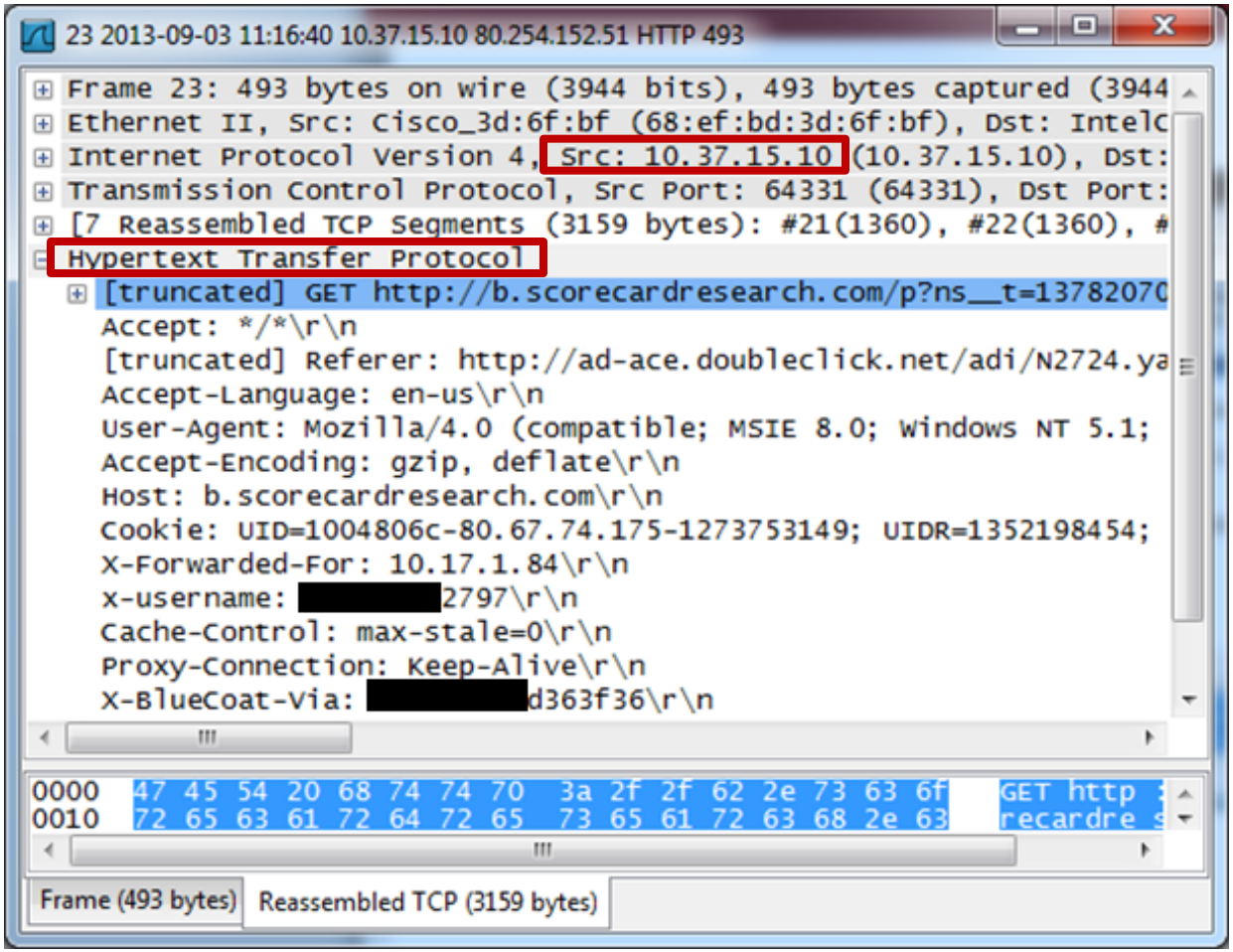

23 2013-09-03 11:16:40 10.37.15.10 80.254.152.51 HTTP 493
Frame 23: 493 bytes on wire (3944 bits), 493 bytes captured (3944
Ethernet II, Src: Cisco_3d:6f:bf (68:ef:bd:3d:6f:bf), Dst: IntelC
Internet Protocol Version 4, Src: 10.37.15.10 (10.37.15.10), Dst:
Transmission Control Protocol, Src Port: 64331 (64331), Dst Port:
[7 Reassembled TCP Segments (3159 bytes): #21(1360), #22(1360), #
Hypertext Transfer Protocol
[truncated] GET http://b.scorecardresearch.com/p?ns__t=1378207C
Accept: */*\r\n
[truncated] Referer: http://ad-ace.doubleclick.net/adi/N2724.ya
Accept-Language: en-us\r\n
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1;
Accept-Encoding: gzip, deflate\r\n
Host: b.scorecardresearch.com\r\n
Cookie: UID=1004806c-80.67.74.175-1273753149; UIDR=1352198454;
X-Forwarded-For: 10.17.1.84\r\n
x-username: [redacted]2797\r\n
Cache-Control: max-stale=0\r\n
Proxy-Connection: Keep-Alive\r\n
X-BlueCoat-Via: [redacted]d363f36\r\n
0000 47 45 54 20 68 74 74 70 3a 2f 2f 62 2e 73 63 6f GET http
0010 72 65 63 61 72 64 72 65 73 65 61 72 63 68 2e 63 recardre
Frame (493 bytes)
Reassembled TCP (3159 bytes)

A More Practical Example


23 2013-09-03 11:16:40 10.37.15.10 80.254.152.51 HTTP 493
Frame 23: 493 bytes on wire (3944 bits), 493 bytes captured (3944
Ethernet II, Src: Cisco_3d:6f:bf (68:ef:bd:3d:6f:bf), Dst: IntelC
Internet Protocol Version 4, Src: 10.37.15.10 (10.37.15.10), Dst:
Transmission Control Protocol, Src Port: 64331 (64331), Dst Port:
[7 Reassembled TCP Segments (3159 bytes): #21(1360), #22(1360), #
Hypertext Transfer Protocol
[truncated] GET http://b.scorecardresearch.com/p?ns__t=1378207C
Accept: */*\r\n
[truncated] Referer: http://ad-ace.doubleclick.net/adi/N2724.ya
Accept-Language: en-us\r\n
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1;
Accept-Encoding: gzip, deflate\r\n
Host: b.scorecardresearch.com\r\n
Cookie: UID=1004806c-80.67.74.175-1273753149; UIDR=1352198454;
X-Forwarded-For: 10.17.1.84\r\n
x-username: [redacted]2797\r\n
Cache-Control: max-stale=0\r\n
Proxy-Connection: Keep-Alive\r\n
X-BlueCoat-Via: [redacted]d363f36\r\n
0000 47 45 54 20 68 74 74 70 3a 2f 2f 62 2e 73 63 6f GET http :
0010 72 65 63 61 72 64 72 65 73 65 61 72 63 68 2e 63 recardre s
Frame (493 bytes)
Reassembled TCP (3159 bytes)

Our Proxy IP

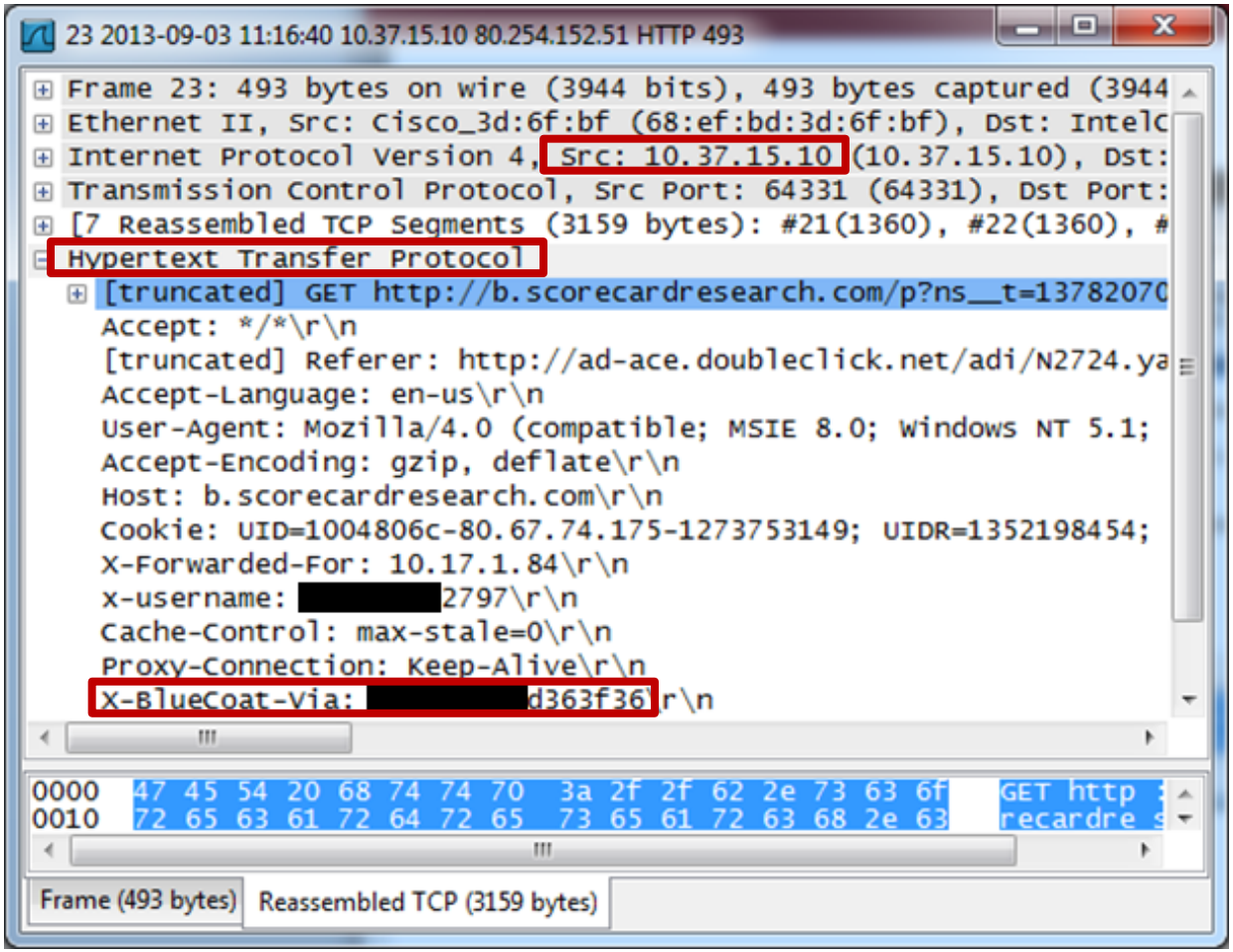
A More Practical Example


23 2013-09-03 11:16:40 10.37.15.10 80.254.152.51 HTTP 493
Frame 23: 493 bytes on wire (3944 bits), 493 bytes captured (3944
Ethernet II, Src: Cisco_3d:6f:bf (68:ef:bd:3d:6f:bf), Dst: IntelC
Internet Protocol Version 4, Src: 10.37.15.10 (10.37.15.10), Dst:
Transmission Control Protocol, Src Port: 64331 (64331), Dst Port:
[7 Reassembled TCP Segments (3159 bytes): #21(1360), #22(1360), #
Hypertext Transfer Protocol
[truncated] GET http://b.scorecardresearch.com/p?ns__t=1378207C
Accept: */*\r\n
[truncated] Referer: http://ad-ace.doubleclick.net/adi/N2724.ya
Accept-Language: en-us\r\n
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1;
Accept-Encoding: gzip, deflate\r\n
Host: b.scorecardresearch.com\r\n
Cookie: UID=1004806c-80.67.74.175-1273753149; UIDR=1352198454;
X-Forwarded-For: 10.17.1.84\r\n
x-username: [redacted]2797\r\n
Cache-Control: max-stale=0\r\n
Proxy-Connection: Keep-Alive\r\n
X-BlueCoat-Via: [redacted]d363f36\r\n
0000 47 45 54 20 68 74 74 70 3a 2f 2f 62 2e 73 63 6f GET http
0010 72 65 63 61 72 64 72 65 73 65 61 72 63 68 2e 63 recardre
Frame (493 bytes)
Reassembled TCP (3159 bytes)

Our Proxy IP

HTTP Dissector

A More Practical Example

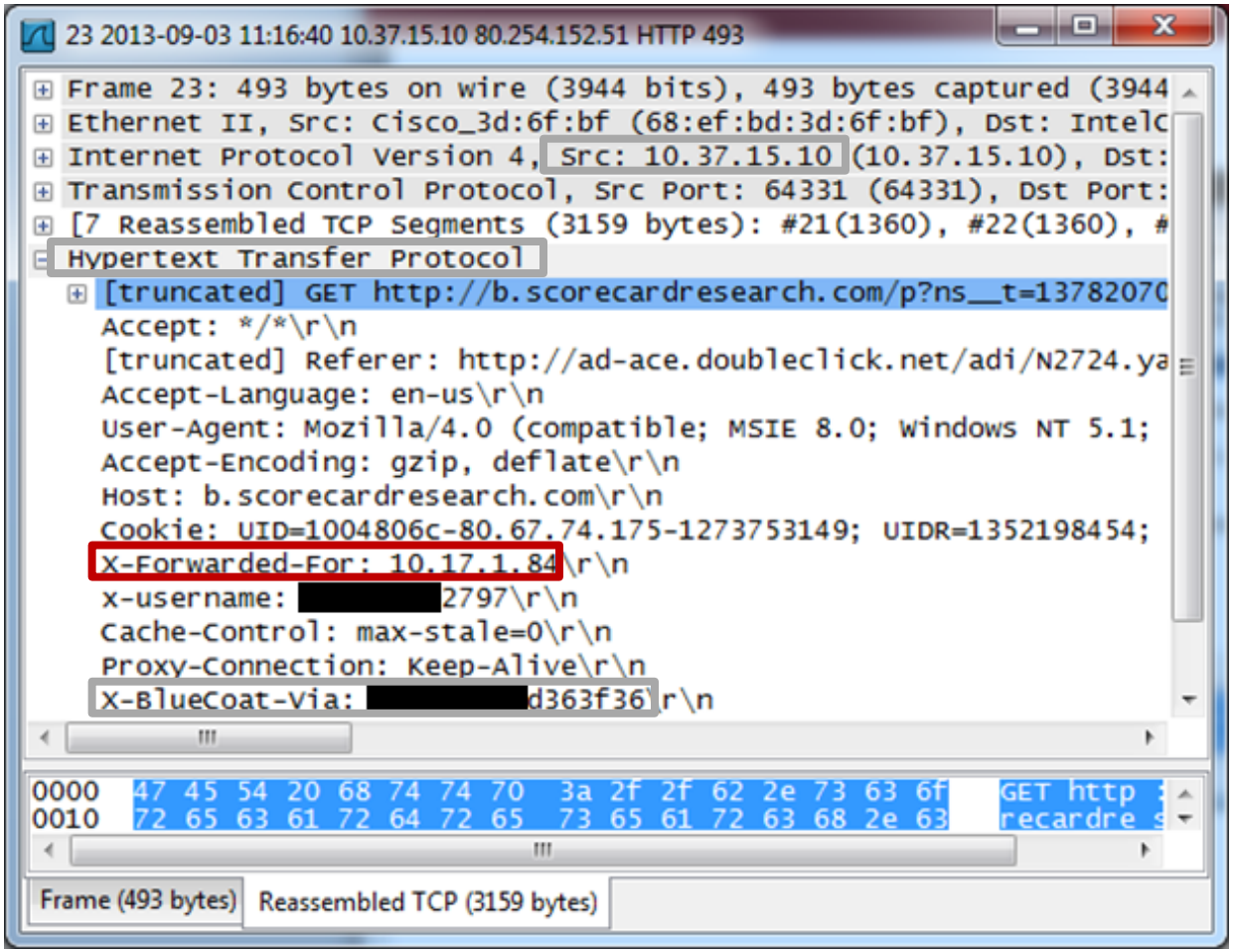

23 2013-09-03 11:16:40 10.37.15.10 80.254.152.51 HTTP 493
Frame 23: 493 bytes on wire (3944 bits), 493 bytes captured (3944
Ethernet II, Src: Cisco_3d:6f:bf (68:ef:bd:3d:6f:bf), Dst: IntelC
Internet Protocol Version 4, Src: 10.37.15.10 (10.37.15.10), Dst:
Transmission Control Protocol, Src Port: 64331 (64331), Dst Port:
[7 Reassembled TCP Segments (3159 bytes): #21(1360), #22(1360), #
Hypertext Transfer Protocol
[truncated] GET http://b.scorecardresearch.com/p?ns__t=1378207C
Accept: */*\r\n
[truncated] Referer: http://ad-ace.doubleclick.net/adi/N2724.ya
Accept-Language: en-us\r\n
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1;
Accept-Encoding: gzip, deflate\r\n
Host: b.scorecardresearch.com\r\n
Cookie: UID=1004806c-80.67.74.175-1273753149; UIDR=1352198454;
X-Forwarded-For: 10.17.1.84\r\n
x-username: [redacted]2797\r\n
Cache-Control: max-stale=0\r\n
Proxy-Connection: Keep-Alive\r\n
X-BlueCoat-Via: [redacted]d363f361r\n
0000 47 45 54 20 68 74 74 70 3a 2f 2f 62 2e 73 63 6f GET http :
0010 72 65 63 61 72 64 72 65 73 65 61 72 63 68 2e 63 recardre s
Frame (493 bytes)
Reassembled TCP (3159 bytes)

Our Proxy IP

HTTP Dissector

Proxy Information

A More Practical Example


23 2013-09-03 11:16:40 10.37.15.10 80.254.152.51 HTTP 493
Frame 23: 493 bytes on wire (3944 bits), 493 bytes captured (3944
Ethernet II, Src: Cisco_3d:6f:bf (68:ef:bd:3d:6f:bf), Dst: IntelC
Internet Protocol Version 4, Src: 10.37.15.10 (10.37.15.10), Dst:
Transmission Control Protocol, Src Port: 64331 (64331), Dst Port:
[7 Reassembled TCP Segments (3159 bytes): #21(1360), #22(1360), #
Hypertext Transfer Protocol
[truncated] GET http://b.scorecardresearch.com/p?ns__t=1378207C
Accept: */*\r\n
[truncated] Referer: http://ad-ace.doubleclick.net/adi/N2724.ya
Accept-Language: en-us\r\n
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1;
Accept-Encoding: gzip, deflate\r\n
Host: b.scorecardresearch.com\r\n
Cookie: UID=1004806c-80.67.74.175-1273753149; UIDR=1352198454;
X-Forwarded-For: 10.17.1.84\r\n
x-username: [redacted]2797\r\n
Cache-Control: max-stale=0\r\n
Proxy-Connection: Keep-Alive\r\n
X-BlueCoat-Via: [redacted]d363f36\r\n
0000 47 45 54 20 68 74 74 70 3a 2f 2f 62 2e 73 63 6f GET http :
0010 72 65 63 61 72 64 72 65 73 65 61 72 63 68 2e 63 recardre s
Frame (493 bytes)
Reassembled TCP (3159 bytes)

Our Proxy IP

HTTP Dissector

Bingo!

Proxy Information

A More Practical Example

Scenario: You are tasked with monitoring a network for HTTP beacons to known network based indicators (NBIs). However, due to the configuration of the network, all IP traffic appears to be from a proxy.

A More Practical Example

Scenario: You are tasked with monitoring a network for HTTP beacons to known network based indicators (NBIs). However, due to the configuration of the network, all IP traffic appears to be from a proxy.

http.request

A More Practical Example

Scenario: You are tasked with monitoring a network for HTTP beacons to known network based indicators (NBIs). However, due to the configuration of the network, all IP traffic appears to be from a proxy.

http.request

http.host

A More Practical Example

Scenario: You are tasked with monitoring a network for HTTP beacons to known network based indicators (NBIs). However, due to the configuration of the network, all IP traffic appears to be from a proxy.

http.request

http.host

http.x_forwarded_for

A More Practical Example

Scenario: You are tasked with monitoring a network for HTTP beacons to known network based indicators (NBIs). However, due to the configuration of the network, all IP traffic appears to be from a proxy.

http.request

http.host

http.x_forwarded_for

```
--Init tap
tap =        Listener.new("http", "http.request")

--Define fields
host =       Field.new("http.host")
org_host =   Field.new("http.x_forwarded_for")
```

A More Practical Example

Scenario: You are tasked with monitoring a network for HTTP beacons to known network based indicators (NBIs). However, due to the configuration of the network, all IP traffic appears to be from a proxy.

http.request

http.host

http.x_forwarded_for

```
--Init tap
tap =        Listener.new("http", "http.request")

--Define fields
host =       Field.new("http.host")
org_host =   Field.new("http.x_forwarded_for")

--Define NBIs
nbis = {}
nbis[1] = "google.com"
nbis[2] = "yahoo.com"
nbis[3] = "bing.com"
```

A More Practical Example

```
function tap.packet(pinfo,tvb,tapdata)

    --Iterate through the NBIs
    for id, nbi in pairs(nbis) do

        --Does the HTTP HOST field contain this NBI
        if string.find(tostring(host()), nbi) ~= nil then

            --Is there an x_forwarded_for header
            if(org_host() ~= nil) then

                --Output CSV record
                message(tostring(org_host())..","..nbis[id]..","..tostring(host()))
            end

        end

    end

end
```

A More Practical Example

```
function tap.packet(pinfo,tvb,tapdata)

    --Iterate through the NBIs
    for id, nbi in pairs(nbis) do

        --Does the HTTP HOST field contain this NBI
        if string.find(tostring(host()), nbi) ~= nil then

            --Is there an x_forwarded_for header
            if(org_host() ~= nil) then

                --Output CSV record
                message(tostring(org_host())..","..nbis[id]..","..tostring(host()))
            end

        end

    end

end
```

A More Practical Example

```
function tap.packet(pinfo,tvb,tapdata)

    --Iterate through the NBIs
    for id, nbi in pairs(nbis) do

        --Does the HTTP HOST field contain this NBI
        if string.find(tostring(host()), nbi) ~= nil then

            --Is there an x_forwarded_for header
            if(org_host() ~= nil) then

                --Output CSV record
                message(tostring(org_host())..","..nbis[id]..","..tostring(host()))
            end

        end

    end

end
```

A More Practical Example

```
function tap.packet(pinfo,tvb,tapdata)

    --Iterate through the NBIs
    for id, nbi in pairs(nbis) do

        --Does the HTTP HOST field contain this NBI
        if string.find(tostring(host()), nbi) ~= nil then

            --Is there an x_forwarded_for header
            if(org_host() ~= nil) then

                --Output CSV record
                message(tostring(org_host())..","..nbis[id]..","..tostring(host()))
            end

        end

    end

end
```

A More Practical Example

```
function tap.packet(pinfo,tvb,tapdata)

    --Iterate through the NBIs
    for id, nbi in pairs(nbis) do

        --Does the HTTP HOST field contain this NBI
        if string.find(tostring(host()), nbi) ~= nil then

            --Is there an x_forwarded_for header
            if(org_host() ~= nil) then

                --Output CSV record
                message(tostring(org_host())..","..nbis[id]..","..tostring(host()))
            end

        end

    end

end
```

A More Practical Example

```
function tap.packet(pinfo,tvb,tapdata)

    --Iterate through the NBIs
    for id, nbi in pairs(nbis) do

        --Does the HTTP HOST field contain this NBI
        if string.find(tostring(host()), nbi) ~= nil then

            --Is there an x_forwarded_for header
            if(org_host() ~= nil) then

                --Output CSV record
                message(tostring(org_host())..","..nbis[id]..","..tostring(host()))
            end

        end

    end

end
```

How'd We Do?

```
10.4.14.155,yahoo.com,beap.adx.yahoo.com
10.4.14.155,yahoo.com,beap.adx.yahoo.com
10.4.14.155,yahoo.com,beap.adx.yahoo.com
10.8.43.118,google.com,apis.google.com
10.6.31.121,yahoo.com,csc.beap.bc.yahoo.com
10.6.31.121,yahoo.com,csc.beap.bc.yahoo.com
10.6.31.121,yahoo.com,csc.beap.bc.yahoo.com
10.24.22.80,google.com,sites.google.com
10.24.22.80,google.com,sites.google.com
10.24.22.80,google.com,sites.google.com
10.5.34.191,yahoo.com,csc.beap.bc.yahoo.com
10.33.138.71,google.com,mail.google.com:443
10.30.10.32,google.com,plus.google.com
10.3.54.112,google.com,safebrowsing.clients.google.com
10.3.54.112,google.com,safebrowsing.clients.google.com
10.3.54.112,google.com,safebrowsing.clients.google.com
10.9.8.228,google.com,mts0.google.com
10.36.23.100,yahoo.com,search.yahoo.com
10.31.3.173,google.com,safebrowsing.clients.google.com
10.67.22.74,yahoo.com,br.yahoo.com
10.3.20.76,google.com,filetransferenabled.mail.google.com
10.3.20.76,google.com,filetransferenabled.mail.google.com
10.3.20.76,google.com,filetransferenabled.mail.google.com
10.3.20.76,google.com,safebrowsing.google.com
10.23.1.188,yahoo.com,sports.yahoo.com
```

Additional Use Cases

- Profiling Network Traffic
  - “Am I seeing everything I should be seeing?”
  - Source networks
  - Protocols
  - VLAN
  - Proxies
- Human Readability & Reporting
  - Extend Wireshark’s dissection capabilities
  - Example: SMB Traffic

SMB Example

```
--Init Listener
tap =          Listener.new("smb")

--Define fields
timestamp    = Field.new("frame.time_epoch")
response     = Field.new("smb.flags.response")
command      = Field.new("smb.trans2.cmd")
pattern      = Field.new("smb.search_pattern")
file         = Field.new("smb.file")
ct           = Field.new("smb.create.time")
```

SMB Example

```
D:\Projects\Wireshark Lua>tshark -q -X lua_script:SMBer.lua -r smb.pcap
1382712061.803373000 - List Directory - \SR10027\*
1382712062.821843000 - List Directory - \SR10027\SR10027\*
1382712063.831917000 - List Directory - \SR10027\SR10027\Import\*
1382712074.557023000 - List Directory - \Temp\T13341.exe
1382712076.018359000 - List Directory - \Temp\T13341.exe
1382712085.347762000 - List Directory - \Temp
1382712085.450561000 - List Directory - \Temp\*
1382712090.117110000 - List Directory - \Temp\T11006 v11.exe
1382712098.212489000 - List Directory - \Temp\T10021v3.1.exe
1382712098.793634000 - List Directory - \Temp\*
1382712104.578539000 - List Directory - \Temp\Setup.exe
1382712104.941233000 - List Directory - \Temp
1382712105.252853000 - List Directory - \Temp\*
1382712140.774069000 - List Directory - \Temp\T11028.exe

D:\Projects\Wireshark Lua>
```

SMB Example

```
-------------------------------------------------
1382712105.252853000 - List Directory - \Temp\*
-------------------------------------------------
.
\Temp\BKOFFICE\I386\MSOADMIN.EXE
\Handbuch\AVK
InternetSecurity
2006.pdf
\Install\data\bzhw8.zfs
\Temp\program
files\Microsoft
BizTalk
Server\Setup\BTSsetupDB.exe
\Temp\COMANDOS\VIDEO\H_Afri.avi
\Temp\EXE\CFS3.EXE
\Temp\Engsetup\engsetup.exe
\Temp\data\Data\Bikes.dat
\Temp
\Temp\.svn
\Temp\Diagnostics\LANGMasterDoctor.exe
-------------------------------------------------
1382712140.774069000 - List Directory - \Temp\T11028.exe
-------------------------------------------------
\SETUP.EXE

D:\Projects\Wireshark Lua>
```

Going Deeper

- Creating Dissectors
  - Perform Custom Analysis
  - Tracking Malicious Traffic
    - Malware Beacons
    - Command and Control Activity
  - Limitations
    - Lack of native Lua functionality
    - May require libraries for full analysis

Summary

- Don’t reinvent the wheel
- Use Lua to extend the capabilities of Wireshark
  - Create listeners to define what traffic you see
  - Create fields to define which values you access
  - Script away:
    - Access “buried” information
    - Trigger custom actions from specific conditions
    - Rewrite output to more useful or readable formats
- Dissector doesn’t exist? Create it!
  - Quickly analyze custom traffic
    - Malware beacons or command and control traffic

Resources

- The Wireshark Wiki
  - http://wiki.wireshark.org/Lua
  - http://wiki.wireshark.org/LuaAPI
- Wireshark Docs
  - http://www.wireshark.org/docs/wsug_html_chunked/wsluarm.html
- Wireshark Development Community
  - http://wiki.wireshark.org/Development
- Lua Docs
  - http://www.lua.org/docs.html

Thank You

- Nick Pelletier
  - Email: nick.pelletier@mandiant.com
  - Twitter: @nrpelletier